REMARKS

This communication is being filed in response to the Office Action having a mailing date of August 24, 2006. The claims are amended as shown. New claims 24-26 are added. No new matter has been added. With this filing, claims 1-26 are pending in the application.

I. Foreign priority

The Office Action acknowledged the claim to foreign priority, but indicated that a replacement certified copy of the priority document needs to be filed/refiled. Accordingly, the replacement certified copy of the priority document is included herewith. With this filing, the claim to foreign priority is perfected.

II. Rejections under 35 U.S.C. § 101

The Office Action rejected claims 1-4 and 6-12 under 35 U.S.C. § 101 for allegedly being directed towards non-statutory subject matter. Specifically, the Office Action stated that "Merely identifying a value indicative of the number of transitions in a circuit does not produce a tangible result."

This rejection is respectfully traversed. It is respectfully submitted that claims 1-4 and 6-12 do in fact meet the requirements of 35 U.S.C. § 101 by producing a tangible result. These claims recite more than just "merely identifying a value indicative of the number of transitions in a circuit." For example, claim 1 as previously presented recited *inter alia* "estimating the power consumption," and claim 10 as previously presented recited *inter alia* "means for estimating the power consumption." Estimation of the power consumption is in fact a tangible result—a person skilled in the art would readily recognize that the estimation of power consumption (of a digital circuit) is a useful result that is important for circuit design, circuit troubleshooting, circuit performance evaluation, and so forth.

However, to facilitate prosecution, claims 1 and 10 are amended as shown to further definitively comply with the requirements of 35 U.S.C. § 101. Specifically, claim 1 is amended to recite, "estimating the power consumption of said digital circuit" to make it clear

that the power estimation is not being performed conceptually, but is being performed in a tangible way for a tangible digital circuit. Moreover, claim 1 is further amended to recite that said value indicative of the number of transitions is used (rather than being "usable") to determine the power consumption, thereby specifically reciting a useful use and tangible result of the claimed "value."

Claim 10 is amended to recite "means for estimating the power consumption of said digital circuit using the acquired number of transitions." Again, this amendment clarifies the estimation of power consumption is for a tangible digital circuit, and further that the acquired number of transitions is in fact used by said estimating.

It is believed that the above-amended language to claims 1 and 10 provide further compliance with the requirements of 35 U.S.C. § 101. However, if the Examiner feels that such language is still lacking, the Examiner is kindly requested to telephone the undersigned attorney to discuss proposed alternative/additional language. It is hoped that such a telephone conference would result in positive resolution of this issue and that any proposed alternative/additional language can be entered by way of Examiner's amendment so as to expedite prosecution to allowance.

III. Claim interpretation

The Office Action rejected claims 1-23 under 35 U.S.C. § 102(b) as being anticipated by a whitepaper entitled "Application of Toggle-Based Power Estimation to Module Characterization," authored by Jochen *et al.* and submitted to the Oldenburg Research and Development Institute for Information Technology Tools and Systems in February 1997. ("*Jochen*"). The applicants disagree with the interpretion (on page 2 of the Office Action) that the phrase "*emulated at hardware level*" as implemented in a system "*configured in the form of a general-purpose digital computer which, appropriately programmed, implements the above mentioned process*," is taught *Jochen.*

It is respectfully submitted that there is a fundamental difference between the operation of the cited reference and the present claims. In order to help the Examiner appreciate certain distinctions between the pending claims and the subject matter of the cited reference, the

applicants will first discuss disclosed embodiments of the present application in comparison to *Jochen*. The present discussion of the differences between the disclosed embodiments and *Jochen* do not define the scope or interpretation of any of the claims. Instead, such discussed differences are intended to merely help the Examiner appreciate important claim distinctions discussed thereafter. The Examiner's specific objections and rejections will be discussed after the helpful discussion of disclosed embodiments.

A. Difference between *simulation* and *emulation*

Simulation, which as disclosed in *Jochen*, refers to using a programmed computer as a model for another system. That is, when a computer is being used as a simulator, the "real" system is not used at all. In a typical simulator, a set of user data is predefined and then presented to a highly customized computer program. The computer program is the "simulator," and it operates on the input data to produce a result. Each separate execution of the computer program (the "simulator") is a new and separate simulation. Further, every time a new simulation is started, it typically continues until completion. A simulator may operate in a loop, but the simulator never varies from its dedicated role as a model for another whole system. One advantage of a simulator is that every aspect of an entire system may be controlled. One disadvantage of a simulator is that every aspect of an entire system must be controlled, and this control often leads to unreasonably large time and system resource requirements.

On the other hand, emulation, as it is employed in the present application, refers to using a programmed computer as a temporary substitute for one component inside a system. Most typically, and as employed in some embodiments discussed in the present application, the emulator is substituted in the place of a system's processor. That is, the "real" processor designed into the system is temporarily replaced with the computer and its emulation program (the "emulator"), and the whole system operates as if the original processor was still present. In an "emulation" (*i.e.*, a system that employs an emulator), there is no requirement that a user predefine a set of data. The emulation begins, and input data is fed into the system by its normal processes. The system operates as if the original processor was still present. The advantage of

an emulator in such a system is that it provides dynamic, real-time access to the system from the perspective of the processor.

B. Examples of *simulation* versus *emulation*

The distinction between "simulation" and "emulation" will be further described by way of an example. Consider a USB webcam in which a CMOS camera is connected to a microprocessor. A simulator would require the developer to predefine the data produced by the CMOS camera and presented to the microprocessor, and the simulator would then "execute" the operations specified in the program code of the microprocessor (simulating the behavior of the microprocessor), and output data as if it was writing to a USB interface. That is, the simulator would be a model for the entire USB webcam. A user could predefine any number of sets of CMOS camera data, and then execute a separate "simulation" for each set.

On the other hand, an emulator would replace only the microprocessor, and it would read image data directly from the CMOS camera. Further, it would write the acquired image data to the USB interface. The user would not need to predefine any data, and when the "emulation" began, it could operate continuously as a stream of CMOS camera data was received from the real CMOS camera.

At a higher level, one component is said to "emulate" another when it performs in exactly the same way, though perhaps not at the same speed. An emulation can be used as a replacement for a part of a system, whereas a simulation is used to analyze a whole system and make predictions about it. In this context, a simulator has no interaction with external hardware. A simulator reads the stimulus from an input file and stores the results in an output file.

C. The present application employs an *emulator*; the cited reference describes a *simulator*

The present application employs *emulation*, and the cited reference employs *simulation*. Further, in the context of the present application, there is no motivation to combine or interchange one technology for the other. In fact, page 1, line 9 through page 3, line 19 of the present application expressly teach the disadvantages/undesirability of simulation, and then discusses the features and benefits of emulation. On the contrary, the cited reference teaches

only simulation. The abstract of the cited reference article recites a "methodology ... embodied in our Toggle Power Simulator (TPS)," and the remainder of the article consistently recites the simulation-concept of TPS. The cited reference article has no express or implied discussion of emulation, and there is no fair reading of the article that suggests emulation.

D. The cited reference does not operate at *hardware level*

The Office Action states that emulating at "hardware level" is analogous to "simulating the circuit at gate-level or RT-level using software that runs on a computer." The applicants respectfully disagree with this analogy/assertion. As described, *inter alia*, at page 4, lines 10-14 of the present application and further described above, embodiments of the present invention employ hardware emulators that operate "*in situ*." That is, the hardware emulator is temporarily placed in the original location of the component that it replaces in the system, *e.g.*, at hardware level. In contrast, *Jochen* describes a simulation of a "hierarchy of abstraction levels," which is not the same as emulation at the hardware level. Jochen does describe "gate-level" and "RT-level" models, but only in the context of estimations provided for use during simulations. *Jochen* does not disclose, teach, or suggest, "emulating at said hardware level," as recited in independent claim 1, for example.

Stated another way, the prior art discloses software simulators for gathering activity data, whereas the present invention proposes using an emulator for analysis of real-time circuit activity data at "hardware level." This hardware level analysis may be achieved by changing the behavior of the hardware emulator, and alteration of the original hardware system is not necessary. In fact, disclosed embodiments of the invention propose to achieve analysis data by changing the libraries of the hardware-level emulator. Some changes include the addition of special modules B, which may, for example, be connected to the output pins of "any cell whatsoever of a digital circuit" G. In this way, the modules B are able to gather the number of transitions performed by the associated cell G.

Therefore, as used in the present specification, the phrase "*that the system may be configured in the form of a general-purpose digital computer which, appropriately programmed, implements the above-mentioned process*" refers to an emulator based on a microprocessor, in

which a program code portion could realize the mentioned modules B. In contrast, section 2 ("Concept of Simulation") and Figure 1 ("Simulation-concept of the TPS") of the cited reference clearly show that the *Jochen* is exclusively limited to software simulation.

From the above discussion, it may be observed that the terms, "simulate" and "emulate," are not the same, and therefore are not interchangeable. Further, it may be observed that emulating at "hardware level" is not fairly and not reasonably analogous to "simulating the circuit at gate-level or RT-level using software that runs on a computer." Each of these two technologies performs different functions in different ways. These differences are recognizable and understood by those skilled in the art.

It is hoped that the above discussion of non-limiting embodiments helps distinguish "software simulation" from "emulation at hardware level" (and systems that are "simulated" from systems that "employ emulators"). In light of the above discussion, it is respectfully submitted that the cited reference does not anticipate any of the present claims. However, to further facilitate movement of the present claims toward allowance, the specific rejections cited by the Examiner will next be discussed in more detail below.

IV. Discussion of the claims

A. Independent claims 1, 10, and 13

Amended claims 1, 10, and 13 are allowable for at least the reason that *Jochen* does not disclose, teach or suggest the feature of "employing a hardware emulator." As described more fully above, hardware emulation and software simulation are distinct and separate technologies. Claims 1, 10, and 13 employ a hardware emulator, and further, their respective new dependent claims 24-26 recite (using varying language) the feature of being able to "handle power estimation also in those cases which cannot [be] handled by ... software simulation," as described in the specification at page 8, lines 10-11. Therefore, because *Jochen* is exclusively limited to software simulation, claims 1, 10, and 13 (and their dependent claims, such as claims 24-26) are allowable.

Amended claims 1, 10, and 13 are further allowable for at least the reason that *Jochen* does not disclose, teach or suggest the feature of "additional element(s) being able to

detect, during emulation." The Office Action has interpreted the "toggle count mechanism employed by the prior art" as analogous to "attaching an additional element to output of the functional element." However, and as explained above, the toggle count mechanism of *Jochen* does not detect during emulation. Rather, Jochen's count of transitions, $N_i(T)$, is entirely based on simulated software input. Section 2 ("Concept of Simulation"), paragraph 4 of *Jochen* recites "information can directly be generated during a logic simulation invoking the toggle count mechanism of the ES2 design kit, or by analyzing the VCD-files" (emphasis added). Neither of the two techniques of *Jochen* contemplates "additional element(s) being able to detect, during emulation," and so, *Jochen* does not anticipate independent claims 1, 10, and 13.

B. Dependent claims 3, 11, and 14

Dependent claims 3, 11, and 14 recite, *inter alia*, "a fraction of time in which a state of the associated functional element is stable." *Jochen* does not disclose, teach, or suggest this feature. The Office Action directs attention to *Jochen's* equations 2-3, but *Jochen's* equations are limited merely to counting the number of transitions in a given time interval, $N_i(T)$ and $N(T)$. Counting a number of transitions in a given time interval is not analogous to "a fraction of time in which a state of the associated functional element is stable," and the present application clearly distinguishes the two values. For example, page 6, lines 3-11 of the present application describe an embodiment of the invention in which three information elements are collected. The first element is a "time interval," the second is a "number of transitions performed," and the third is "the fraction of time in which the state is stable ... which can be expressed as a percentage." Although *Jochen's* equations might hypothetically suggest the first two elements, *Jochen's* equations clearly do not in any way contemplate "the fraction of time in which the state is stable." For at least the reason that *Jochen* does not disclose, teach, or suggest the limitation of "the fraction of time in which the state is stable," claims 3, 11, and 14 are allowable.

B. Dependent claims 4, 8, 9, 12, and 16-23

Each of the dependent claims 4, 8, 9, 12, and 16-23 recite limitations that one skilled in the art will readily recognize as related to hardware. Conversely, the cited *Jochen* reference is exclusively limited to software. Claim 4 recites, *inter alia*, "hardware events monitored by logic analyzers active on the additional elements." Claim 19 recites, *inter alia*, "acquiring said value in real time." Claims 8, 9, 12, and 16 recite, *inter alia*, "emulating at the hardware level." Claims 17, 20, and 22 recite, *inter alia*, formation "of the digital circuit." And claims 18, 21, and 23 recites, *inter alia*, "during [emulation or operation] of the digital circuit."

As more fully described above, *Jochen* does not disclose, teach, or suggest any association with hardware. The text of the *Jochen* reference describes only software simulation and discloses only software simulation and design tools (TPS, HSPICE, SYNOPSIS, ISCAS'85, and CADENCE for example). Accordingly, for at least the reason that the software simulations disclosed by *Jochen* cannot envisage use of "hardware events," "logic analyzers," "emulating at hardware level," or formation/emulation of "digital circuitry," claims 4, 8, 9, 12, and 16-23 are allowable.

V. Conclusion

Overall, the cited reference does not disclose, teach, or suggest what is recited in the independent claims. Thus, given the above amendments and accompanying remarks, the independent claims are now in condition for allowance. The dependent claims that depend directly or indirectly on these independent claims are likewise allowable based on at least the same reasons and based on the recitations contained in each dependent claim.

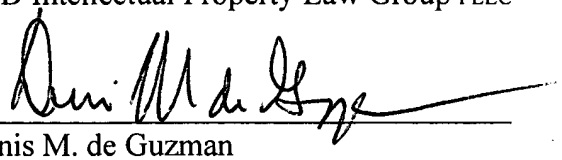
If the undersigned attorney has overlooked a teaching in any of the cited references that is relevant to the allowability of the claims, the Examiner is requested to specifically point out where such teaching may be found. Further, if there are any informalities or questions that can be addressed via telephone, the Examiner is encouraged to contact the undersigned attorney at (206) 622-4900.

The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC



Dennis M. de Guzman
Registration No. 41,702

DMD:jr

701 Fifth Avenue, Suite 5400
Seattle, Washington 98104
Phone: (206) 622-4900
Fax: (206) 682-6031

851763.420 / 862321_1.DOC